CLAIMS

What is claimed as new and desired to be protected by Letters Patent of the United States is:

1. A method for decreasing the latency between an instruction cache and a pipeline processor having a plurality of parallel execution stages, each execution stage having a decode stage and an instruction queue for sequentially processing instructions being processed by said processor, comprising:

determining whether said decode stage and instruction queue do not have valid data; and

inserting instructions from said instruction cache in parallel to said decode stage and instruction queue when said decode stage and instruction queue contain invalid data.

2. A method for decreasing the latency between an instruction cache and said pipeline processor according to claim 1 further comprising:

processing said cache instructions from said cache sequentially through said decode stage and instruction queue when valid data exists in said instruction queue.

3. A method for processing instructions in a pipelined processor having a series of pipelined stages which reduces latency between an instruction queue and a pipeline processor comprising:

serially fetching a plurality of instructions to be executed in said pipeline processor from a cache memory;

decoding each of said fetched instruction addresses in a first stage of said pipeline processor to determine if an execution branch is to be taken;

loading said instruction into said instruction queue at the same time said instruction is being loaded in said decoder when said instruction queue and decoder are empty;

sequentially loading said instruction into said instruction queue from said decoder when said instruction queue and said decoder are not empty; and

shifting the contents of an instruction queue to produce an instruction from said instruction queue for processing in subsequent pipeline stages.

4. The method for processing instructions in a pipelined processor according to claim 3 wherein said decoder identifies said instructions loaded in said queue at the same time as said instructions which are loaded in said decoder as valid or invalid during a subsequent cycle of said pipeline processor if an execution branch is not taken.

5. The method for processing instructions in a pipelined processor according to claim 3 wherein said instruction queue contents are shifted left to an output port connected to plural pipelined processor stages.

6. A method for executing instructions in a pipelined processor comprising:

sequentially fetching the addresses of instructions to be executed by said pipelined processor;

determining if said instructions are stored in a cache memory;

determining whether a decode stage and location of an instruction queue stage of said pipe line processor is empty;

loading said instruction from said cache memory in said decode stage and said instruction queue in parallel when said stages are empty; and

sequentially reading out said instruction queue instructions for execution in said pipelined processor.

7. The method for executing instructions in a pipelined processor according to claim 6 further comprising:

loading only said decode stage with said instructions when said instruction queue contains valid data, and sequentially transferring said instructions to said instruction queue when a position in said instruction queue is available.

8. The method for executing instructions in a pipelined processor according to claim 7 further comprising identifying said instruction as a branch instruction if said decoder predicts that a branch is being taken from said instruction; and

inhibiting transfer of subsequent instructions from said decoder to said instruction queue.

9. The method for executing instructions in a pipeline processor according to claim 7 further comprising:

fetching an instruction from a main memory when said instruction is not in said cache memory; and

forwarding said instruction to said decode stage for sequential transfer to said instruction queue.

10. The method for executing instructions in a pipeline processor according to claim 6 further comprising:

examining the contents of a location in said queue; and

determining the state of a valid bit in each of said locations whereby the determination of whether said location contains valid data is made.

11. The method for executing instructions in a pipeline processor according to claim 7, wherein said instructions are transferred from said decode stage to said instruction queue each time an instruction is read from said instruction queue.

12. An apparatus for reducing the latency between stages of a pipelined processor comprising:

an instruction cache producing a plurality of instructions for execution by said pipelined processor;

a plurality of decode stages connected to receive a plurality of instructions from said cache;

an instruction queue having a plurality of locations for receiving an instruction and a valid bit; and

a plurality of multiplexers for receiving each of said instructions, an output of a respective decode stage receiving said instructions, and connected to receive a valid bit from a location of said queue as a select signal, said multiplexer connected to supply each of said instruction queue locations with one of said instructions selected from either from said decoder stage or from said cache.

13. The apparatus according to claim 12, wherein said multiplexer receives a shift signal for said instruction queue which shifts the contents of said instruction queue towards an output port of said queue, and which enables said instructions from said decoder to be transferred to said instruction queue.

14. The apparatus according to claim 12 wherein said output port of said queue is connected to a plurality of parallel processing stages.

15. The apparatus according to claim 14 wherein said processing stages execute instructions belonging to one of a load/store operation, arithmetic operation, or a branch target instruction.